HYPERCF: BIDIRECTIONAL INCIDENCE COUNTERFACTUALS FOR HYPERGRAPH NEURAL NETWORKS

Anonymous authors
Paper under double-blind review

ABSTRACT

Hypergraphs model higher-order interactions that pairwise graphs cannot represent, and hypergraph neural networks (HGNNs) are increasingly used to learn over them. As these models enter decision settings, structural counterfactual explanations become important: they ask for a minimal incidence edit under which the prediction would change. Existing hypergraph counterfactual methods are limited to deletion-only interventions on the original incidence structure, so the search fails on many targets. We expand the intervention space to bidirectional incidence editing, allowing a target both to leave incidences and to join existing hyperedges, and present HyperCF, which controls this expansion with a hop-bounded additive candidate space, learning-rate calibration based on deletion gradients with a coupled additive rate, and symmetric redundancy pruning. Out of seven widely adopted benchmarks, HyperCF raises counterfactual success over deletion-only search on six datasets at a two-hop radius and five at a one-hop radius, with gains of up to 40 percentage points, while the mean explanation size remains comparable and often smaller.

1 INTRODUCTION

Many systems are governed by group interactions rather than pairwise links alone: co-authorship teams, biochemical complexes, and multi-item sessions are natural examples Battiston et al. (2020). Hypergraphs represent such relations by letting one hyperedge connect an arbitrary set of nodes Zhou et al. (2006), and hypergraph neural networks (HGNNs) Feng et al. (2019); Yadati et al. (2019); Chien et al. (2022); Kim et al. (2024) are the standard predictors on this representation. When an HGNN assigns a label, a structural counterfactual is a minimal incidence change under which that prediction would differ Wachter et al. (2017); Lucic et al. (2022b).

On pairwise graphs, counterfactual search has already moved beyond deletion-only edits. Methods such as CF-GNNExplainer optimize masks over adjacency Lucic et al. (2022b); ATEX-CF and C2Explainer further allow edge addition as well as deletion Zhang et al. (2026); Ma et al. (2025); COMBINEX jointly edits structure and features Giorgi et al. (2025b). On hypergraphs, however, explainability is still dominated by factual attribution. HyperEX and SHypX identify prediction-supporting substructures Maleki et al. (2023); Su et al. (2024), while the only hypergraph counterfactual method, CF-HyperGNNExplainer Veglianti et al. (2026), searches by deleting incidences or whole hyperedges. Neither variant can add a target node to an existing hyperedge. Deletion-only search is therefore limited to sparsifications of the target's original incidence row, and it fails on many targets.

The root issue is the action space. A hypergraph incidence has two directions: a node can leave a hyperedge it belongs to, or join a hyperedge that already exists nearby. Deletion removes context; addition supplies an existing higher-order group that the target does not yet participate in. The scientific question is therefore:

> *Can bidirectional incidence editing find counterfactuals that deletion-only search fails to generate, and what controls are required to keep the expanded search effective and minimal?*

Expanding the action space introduces three failure modes:

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

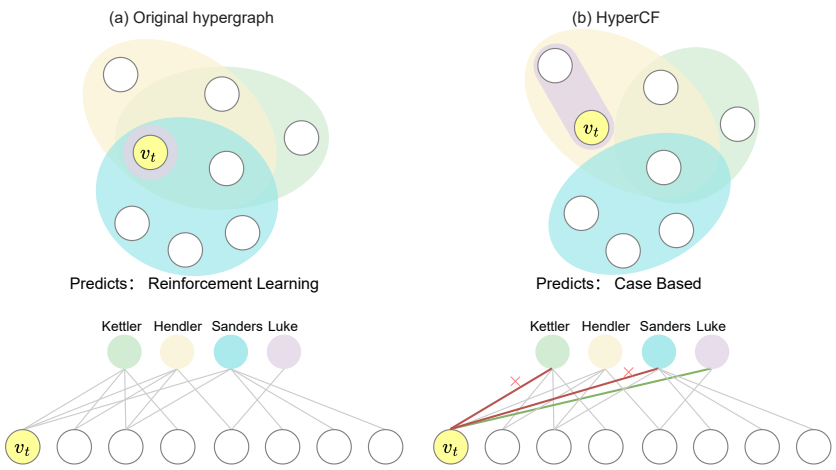


Figure 1: A real Cora-CA case illustrating a counterfactual that deletion-only search cannot find.

- **Candidate explosion.** The full set of hyperedges a node could join is large. Unrestricted candidates drown the gradient signal and include many hyperedges outside the model's original receptive field.
- **Optimization imbalance.** Additive parameters outnumber deletion parameters. Joint learning-rate calibration from the full gradient lets additive noise shrink the rate needed by deletions, so the joint search can fall below the deletion-only baseline.
- **Redundant edits.** A larger action space encourages the optimizer to accumulate unnecessary deletions and additions, inflating explanation size.

Case study Figure 1 shows a real counterfactual from Cora-CA, where nodes are papers and hyperedges are authors. In the original hypergraph, the target paper v_t belongs to the Kettler, Hendler, and Sanders hyperedges, and the HGCN predicts *Reinforcement Learning*. The existing Luke hyperedge shares a neighboring paper with v_t through the local coauthorship structure, but does not initially contain v_t; it is therefore a one-hop additive candidate. Deletion-only search finds no counterfactual for this target. HyperCF removes v_t from the Kettler and Sanders hyperedges, retains its membership in Hendler, and adds it to the Luke hyperedge. This bidirectional incidence edit changes the prediction to *Case Based*. The example shows how joining an existing nearby hyperedge can provide a counterfactual action unavailable to deletion-only search.

HyperCF implements a single soft incidence mask over both directions and attaches one control to each failure mode: a hop-bounded additive candidate set, learning-rate calibration based on deletion gradients with a coupled additive rate, and symmetric redundancy pruning. Bidirectional edits and post-hoc pruning already exist for pairwise edges Zhang et al. (2026); Ma et al. (2025). On hypergraphs the edited unit is a node–hyperedge incidence rather than a pairwise edge, which changes the candidate geometry, the optimization balance, and the meaning of redundancy. HyperCF searches this incidence space in both edit directions.

Our contributions are as follows.

- **Bidirectional incidence counterfactuals for HGNNs.** We formulate hypergraph counterfactual search as an incidence editing problem and show that deletion-only methods are limited to sparsifications of the original incidence row. We then define an action space with deletion and addition in which a target may leave existing hyperedges and join hyperedges that already occur nearby.
- **A controlled search method for the expanded action space.** We propose HyperCF, which enlarges the incidence mask to both edit directions and introduces three controls that make the expansion usable: hop-bounded additive candidates, learning-rate calibration based on deletion gradients with a coupled additive rate, and symmetric redundancy pruning. Together, these mechanisms address

candidate explosion, optimization imbalance, and redundant edits without changing the baseline objective of prediction change and edit size.

- **Empirical evidence of effectiveness and limits.** Across seven public benchmarks, HyperCF improves counterfactual success over deletion-only search on most datasets, while mean explanation sizes do not increase and often decrease. Returned explanations on the largest-gain datasets are dominated by additions, and DBLP-CA isolates the case where additive candidates become redundant or hard to optimize.

2 Related Work

Counterfactual explanations Counterfactual explanations describe a minimal input change that yields a different model outcome Wachter et al. (2017). The idea has been developed for tabular features Tolomei et al. (2017); Tolomei & Silvestri (2021); Mothilal et al. (2020); Poyiadzi et al. (2020), tree ensembles and sequential decisions Lucic et al. (2022a); Chen et al. (2022), and other structured outputs Chen et al. (2025); Giorgi et al. (2025a). The rest of this section turns to structural interventions and keeps the standard prediction change plus distance objective.

Counterfactual and attribution explainers for graphs Attribution methods such as GNNExplainer Ying et al. (2019) and PGExplainer Luo et al. (2020) highlight predictive substructures. CF-GNNExplainer Lucic et al. (2022b) optimizes a continuous mask over the adjacency matrix; RCExplainer Bajaj et al. (2022) adds robustness constraints; COMBINEX Giorgi et al. (2025b) jointly edits edges and node features. Applying these methods to a hypergraph requires an expansion into a graph, which is memory-prohibitive on large hypergraphs such as DBLP-CA Veglianti et al. (2026).

Masks over existing edges leave addition out of reach. ATEX-CF Zhang et al. (2026) unifies edge addition and deletion, draws additive candidates from attack heuristics, and jointly optimizes impact, sparsity, and plausibility. C2Explainer Ma et al. (2025) likewise inserts as well as removes edges, subject to user constraints. Both edit pairwise links. On a hypergraph the intervention changes a node's membership in an existing group, so the candidate set, the balance between deletion and addition, and the meaning of a redundant edit differ. Post-hoc pruning remains useful, as in ATEX-CF, but an incidence prune must reverse the two directions.

Hypergraph learning and explainability HGNNs propagate information over the incidence matrix Feng et al. (2019); Yadati et al. (2019); Chien et al. (2022); Kim et al. (2024). Existing HGNN explainers are largely factual: HyperEX Maleki et al. (2023) scores node-hyperedge pairs, SHypX Su et al. (2024) samples faithful sub-hypergraphs, and interpretable architectures include a self-explaining HGNN Yu et al. (2025) and explainable hypergraph convolution an Bi et al. (2023). None of these generates a counterfactual.

The closest prior work is CF-HyperGNNExplainer Veglianti et al. (2026). Its NHP (Node–Hyperedge Perturbation) variant deletes node-hyperedge incidences and its HP (Hyperedge Perturbation) variant deletes whole hyperedges; neither can add a target node to an existing hyperedge. No hypergraph counterfactual method both deletes and adds incidences. That is the setting of HyperCF.

3 Problem Formulation

3.1 Hypergraphs and Hypergraph Neural Networks

A hypergraph is a pair $\mathcal{G} = (V, E)$ with node set V and hyperedge set E, where each hyperedge $e \in E$ is a non-empty subset of V. We represent $\mathcal{G}$ by its binary *incidence matrix* $\boldsymbol{H} \in \{0,1\}^{N\times M}$, where $\boldsymbol{H}_{ie} = 1$ iff $v_i \in e$. Let $\boldsymbol{W} \in \mathbb{R}^{M\times M}$ be a diagonal matrix of positive hyperedge weights. Let D and B denote the node-degree and hyperedge-degree matrices,

$$D_{ii} = \sum_e \boldsymbol{W}_{ee}\boldsymbol{H}_{ie}, \qquad B_{ee} = \sum_i \boldsymbol{H}_{ie}.$$

For a trained HGNN, one normalized hypergraph convolution layer is

$$X^{(l+1)} = \sigma\big(S(\boldsymbol{H})\, X^{(l)} P^{(l)}\big), \qquad S(\boldsymbol{H}) = D^{-\frac{1}{2}}\boldsymbol{H}\,\boldsymbol{W}\,B^{-1}\boldsymbol{H}^{\top}D^{-\frac{1}{2}}, \tag{1}$$

where $P^{(l)}$ is the layer weight matrix and $\sigma(\cdot)$ denotes a pointwise activation. Our model uses LeakyReLU. We use unit hyperedge weights, $\boldsymbol{W} = \boldsymbol{I}$, in all experiments. A trained L-layer HGNN with a final prediction head maps an incidence structure to a predicted class for each node. Throughout the paper, the predictor parameters, node features, and hyperedge weights are fixed; counterfactual search modifies only $\boldsymbol{H}$. We write $f(\boldsymbol{H})_{v_t}$ for the predicted class of the target.

3.2 Deletion-Only Incidence Counterfactuals

For a target node v_t, let $\hat{y}_t = f(\boldsymbol{H})_{v_t}$ denote its original prediction. A structural counterfactual is an edited incidence matrix $\boldsymbol{H}'$ satisfying

$$f(\boldsymbol{H}')_{v_t} \neq \hat{y}_t.$$

The action space determines which counterfactuals are admissible. In deletion-only search, the target may leave only hyperedges to which it is already incident:

$$\mathcal{D}(v_t) = \{e \in E : \boldsymbol{H}_{v_t e} = 1\}. \tag{2}$$

Thus, the target row of $\boldsymbol{H}'$ can only be a sparsification of its original row. Deletion-only counterfactuals are therefore limited to prediction changes expressible by removing existing memberships.

3.3 Bidirectional Incidence Counterfactuals

An incidence edit has two directions. A target can leave a hyperedge it belongs to, or join a hyperedge that already exists but currently excludes it. We therefore define the full addable set

$$\mathcal{A}(v_t) = \{e \in E : \boldsymbol{H}_{v_t e} = 0\}, \tag{3}$$

and represent an admissible structural edit by a pair (Δ^-, Δ^+) with $\Delta^- \subseteq \mathcal{D}(v_t)$ and $\Delta^+ \subseteq \mathcal{A}(v_t)$. The edited incidence matrix is obtained by removing the incidences in Δ^- and adding those in Δ^+. Deletion-only search is the special case $\Delta^+ = \varnothing$.

The full addable set is too large for direct optimization and includes hyperedges outside the model's original neighborhood. Section 4 therefore searches a hop-bounded subset $\mathcal{A}_h(v_t) \subseteq \mathcal{A}(v_t)$.

4 Method

This section presents HyperCF. Search keeps the objective of prediction change and edit size used by CF-HyperGNNExplainer Veglianti et al. (2026) and enlarges the incidence mask so that a target can both leave existing hyperedges and join hop-bounded ones. We proposal three techniques to make searching in the larger action space feasible and effective: a hop-bounded additive candidate set, learning-rate calibration based on deletion gradients with a coupled additive rate, and symmetric redundancy pruning. They address candidate explosion, optimization imbalance, and redundant edits, respectively.

4.1 Joint deletion and addition

For a target v_t, deletions are taken from $\mathcal{D}(v_t)$ in Eq. 2. Additions are not taken from the full addable set $\mathcal{A}(v_t)$, which is prohibitively large. We define the h-hop incidence neighborhood of v_t as

$$\mathcal{N}_h(v_t) = \{\, u \in V : ((\boldsymbol{H}\boldsymbol{H}^\top)^h)_{v_t u} > 0 \,\}, \tag{4}$$

and the *hop-bounded additive candidates* as

$$\mathcal{A}_h(v_t) = \{\, e \in E : \boldsymbol{H}_{v_t e} = 0 \text{ and } \exists\, u \in \mathcal{N}_h(v_t) \text{ with } \boldsymbol{H}_{ue} = 1 \,\}. \tag{5}$$

When h=1, $\mathcal{A}_h(v_t)$ contains hyperedges that include a node of $\mathcal{N}_h(v_t)$ but not v_t; when h=2, nodes within two incidence hops are eligible, and the set additionally contains hyperedges of nodes two hops away. The target's original h-hop neighborhood is a locality restriction for constructing these candidates. The restriction is not complete: adding the target to a remote hyperedge can itself create a new communication path. It trades search coverage for a bounded candidate space, and we evaluate $h \in \{1, 2\}$ empirically in Section 5.3.

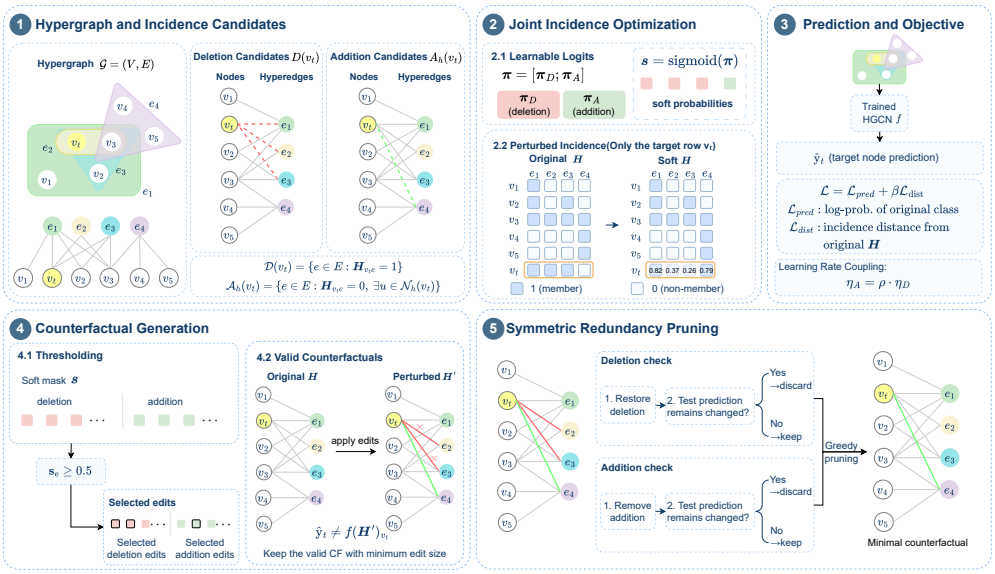


Figure 2: Overview of HyperCF.

A single vector $\boldsymbol{\pi} = [\boldsymbol{\pi}_D; \boldsymbol{\pi}_A] \in \mathbb{R}^{|\mathcal{D}|+|\mathcal{A}_h|}$ of logits parameterizes both directions, with $s_e = \mathrm{sigmoid}(\boldsymbol{\pi}_e)$. The deletion logits are initialized to 4.0, so $\mathrm{sigmoid}(4.0) \approx 0.98$ and almost all incidences are kept. The addition logits are initialized to -4.0, so $\mathrm{sigmoid}(-4.0) \approx 0.018$ and almost no additions are active. Other additive initializations are compared in Table 5. The soft edited incidence equals $\boldsymbol{H}$ except on the target row, where candidate entries take the value s_e; the hard edited incidence $\boldsymbol{H}'$ sets each candidate entry to one iff $s_e \geq 0.5$. Because additions only join hyperedges already present in E, the edited incidence never creates a new hyperedge; it changes only the target's membership row. Hyperedges are the indexed columns of $\boldsymbol{H}$. An addition does not merge a column with another of equal membership, and such a column remains an additive candidate.

For the forward pass we replace $\boldsymbol{H}$ in the propagation operator with $\boldsymbol{H}(\boldsymbol{\pi})$ and recompute the degree matrices on the perturbed structure,

$$S(\boldsymbol{H}(\boldsymbol{\pi})) = \hat{D}^{-\frac{1}{2}} \boldsymbol{H}(\boldsymbol{\pi})\, \hat{B}^{-1} \boldsymbol{H}(\boldsymbol{\pi})^{\top} \hat{D}^{-\frac{1}{2}}, \tag{6}$$

so that normalization stays consistent with the edited connectivity. A column or row with degree zero is omitted from the normalized operator: the corresponding entry of $\hat{B}^{-1}$ or $\hat{D}^{-\frac{1}{2}}$ is set to zero.

We adopt the loss design of CF-HyperGNNExplainer Veglianti et al. (2026) and define

$$\mathcal{L}(\boldsymbol{\pi}) = \mathcal{L}_{\text{pred}}\big(f(\boldsymbol{H}(\boldsymbol{\pi})), \hat{y}_t\big) + \beta\, \mathcal{L}_{\text{dist}}(\boldsymbol{\pi}), \tag{7}$$

where β trades off a prediction change against the edit size. We instantiate $\mathcal{L}_{\text{pred}}$ as the log-probability of the original class under the soft perturbation, $\mathcal{L}_{\text{pred}} = \log p_y(\boldsymbol{\pi})$. Minimizing it lowers confidence in $\hat{y}_t$ and encourages a flip, without choosing a target class. We instantiate $\mathcal{L}_{\text{dist}}$ as the expected edit load on each direction,

$$\mathcal{L}_{\text{dist}}(\boldsymbol{\pi}) = \frac{1}{|\mathcal{D}|} \sum_{e \in \mathcal{D}} (1 - s_e) + \frac{1}{|\mathcal{A}_h|} \sum_{e \in \mathcal{A}_h} s_e, \tag{8}$$

dropping a sum when its candidate set is empty. Both terms lie in $[0, 1]$, so a single β regularizes the two directions on the same scale.

Each epoch thresholds the soft mask at 0.5. A hard structure is valid only if it flips the target prediction. Among valid structures we keep the one with the fewest changed incidences.

4.2 DELETION LEARNING-RATE CALIBRATION

The deletion-only baseline sets a per-node learning rate from the initial prediction gradient:

$$\eta_D = \frac{\tau + \log C}{T - 1} \cdot \frac{1}{\|\nabla_{\boldsymbol{\pi}_D}(\log p_y(\boldsymbol{\pi}))\|^2 + \varepsilon}, \tag{9}$$

where $\tau = 0.1$, C is the number of classes, ε is a small positive constant, and T is the number of optimization epochs per trial. Replacing the deletion parameter set with the concatenation $[\boldsymbol{\pi}_D; \boldsymbol{\pi}_A]$ breaks this calibration: the additive parameters now outnumber the deletion parameters, and because additions start at $\mathrm{sigmoid}(-4.0) \approx 0.018$, their initial gradients are small and noisy relative to their count. A joint gradient norm is then dominated by additive noise, and the calibrated rate shrinks below what the deletion search needs. We therefore calibrate η_D with the additive logits held at their initial state, so that the calibration forward pass matches the deletion-only search. The additive parameters are then updated at a fixed fraction of that rate,

$$\eta_A = \rho\, \eta_D. \tag{10}$$

We use $\rho = 0.1$ as the default. Both rates are computed once at initialization and then held fixed. Coupling η_A to η_D keeps additive updates on the same per-node scale as the deletion search, but smaller. Other ratios are studied in Appendix C.

Each epoch evaluates the soft mask and the thresholded hard structure, and records valid counterfactuals as above. Updates use Nesterov SGD with momentum 0.9 and gradient clipping. Search stops when the perturbation saturates, when gradients vanish with a stable prediction for several consecutive epochs, or when the epoch budget is exhausted. Algorithm 1 summarizes the procedure.

Algorithm 1 HyperCF: bidirectional incidence counterfactual search.

Require: Incidence $\boldsymbol{H}$, trained HGNN f, target node v_t
Construct deletion set $\mathcal{D}(v_t)$ and hop-bounded additive set $\mathcal{A}_h(v_t)$
Initialize $\boldsymbol{\pi}_D, \boldsymbol{\pi}_A$ as defined above; calibrate η_D and set $\eta_A \leftarrow \rho\, \eta_D$
$\boldsymbol{H}'_{\text{best}} \leftarrow \varnothing$
for each optimization step within the prescribed budget **do**
    Compute soft $\boldsymbol{H}(\boldsymbol{\pi})$ and threshold it to obtain $\boldsymbol{H}'$
    if $\boldsymbol{H}'$ flips the prediction **then**
        if $\boldsymbol{H}'_{\text{best}} = \varnothing$ **or** $\boldsymbol{H}'$ has strictly fewer incidence edits **then**
            $\boldsymbol{H}'_{\text{best}} \leftarrow \boldsymbol{H}'$
        end if
    end if
    Update $\boldsymbol{\pi}$ by $\mathcal{L} = \mathcal{L}_{\text{pred}} + \beta\, \mathcal{L}_{\text{dist}}$
end for
if $\boldsymbol{H}'_{\text{best}} = \varnothing$ **then**
    return failure
end if
return $\mathrm{SYMMETRICPRUNE}(\boldsymbol{H}'_{\text{best}})$

4.3 SYMMETRIC REDUNDANCY PRUNING

The distance term averages the edit penalty over a large candidate set, so a solution near the mask threshold can keep deletions and additions that are not required for the flip. Post-hoc pruning of redundant structural edits is standard in graph counterfactuals Zhang et al. (2026). On incidences the two directions reverse differently: undoing a deletion restores a membership, while undoing an addition removes one.

An edit is *redundant if reverting it still leaves the prediction flipped*. We run the greedy twice, once processing deletion reverts first and once processing addition removals first, because the two orders can stop at different subsets, and we return the smaller explanation. Each trial revert is re-evaluated by the model, so the output remains a valid counterfactual and is no larger than either ordering alone. Algorithm 2 states the procedure. The ablation in Appendix D shows that pruning does not change success and mainly removes surplus additions.

Algorithm 2 Symmetric pruning.

Input: Counterfactual $\boldsymbol{H}'$, original $\boldsymbol{H}$, model f, target node v_t
Output: Minimal perturbation Δ^*
1. Compute Δ^-, Δ^+ by comparing $\boldsymbol{H}'$ with $\boldsymbol{H}$; initialize $\Delta^* \leftarrow \Delta^- \cup \Delta^+$
2. Order incidences in Δ^* as Δ^- then Δ^+
3. **for** each incidence e_i in this order:
   (a) $\Delta' \leftarrow \Delta^* \setminus \{e_i\}$, and let $\boldsymbol{H}'$ apply Δ' to $\boldsymbol{H}$ (tentatively remove)
   (b) **if** $f(\boldsymbol{H}')_{v_t} \neq \hat{y}_t$:
      i. $\Delta^* \leftarrow \Delta'$ (keep the smaller perturbation)
4. Repeat Steps 2–3 in the reverse order and **return** the smaller Δ^*

5 Experimental Evaluation

Datasets and base model We evaluate seven public hypergraph benchmarks Chien et al. (2022): the cocitation datasets Cora, Citeseer, and Pubmed; the co-authorship datasets Cora-CA and DBLP-CA; and the label-encoding datasets Mushroom and NTU2012. Dataset statistics are reported in Appendix B. The base model is a three-layer HGCN with hidden dimensions 64 and 32, LeakyReLU, dropout 0.5, and a final linear classification layer. It is trained with Adam for 200 epochs at learning rate 0.01 and weight decay 5×10^{-4}, on the standard AllSet split. After AllSet preprocessing, the cocitation hypergraphs contain many isolated nodes Chien et al. (2022). Those nodes admit no structural edit and are left out of the success-rate denominator.

Protocol For each dataset we sample 1,000 test nodes; NTU2012 uses the full test split, which is smaller. CF-HyperGNNExplainer and HyperCF are evaluated with three random seeds 42, 123, and 456, and Table 1 reports the arithmetic mean. Results for HyperEX, SHypX, and CF-GNNExplainer are taken from CF-HyperGNNExplainer Veglianti et al. (2026). The pruning ablation and the ρ/initialization scans use one random seed. Those scans are reported in Appendix C.

The counterfactual search uses 50 optimization epochs per trial under an incremental-β protocol: β starts at 0, doubles until the first failure, and is then refined by log-bisection within a budget of 30 trials. Updates use Nesterov SGD with momentum 0.9, gradient clipping at norm 2, and hard thresholding at 0.5. The deletion learning rate η_D is calibrated once from Eq. 9 and then held fixed; HyperCF sets $\eta_A = \rho\eta_D$ with default $\rho = 0.1$. The deletion-only baseline is our reproduction of the NHP variant of CF-HyperGNNExplainer under this search procedure, so the comparison holds the base model and optimizer fixed and isolates the edit action space. HyperCF searches deletions from $\mathcal{D}(v_t)$ and additions from $\mathcal{A}_h(v_t)$ at radius $h \in \{1, 2\}$ as independent runs, and applies the symmetric prune. We ran all experiments using NVIDIA A100 GPUs with 40 GB of memory.

Baselines and metrics We compare against CF-HyperGNNExplainer Veglianti et al. (2026), the deletion-only hypergraph counterfactual baseline, and CF-GNNExplainer Lucic et al. (2022b) applied to the star-expanded hypergraph. CF-GNNExplainer cannot run on DBLP-CA because the expansion exceeds memory, matching the baseline study Veglianti et al. (2026). We also include HyperEX Maleki et al. (2023) and SHypX Su et al. (2024), which are factual explainers rather than native counterfactual methods. For each of them we keep the selected factual sub-hypergraph, form the complementary incidence-difference structure, and test whether that structure flips the prediction. These rows are therefore an indirect reading of a factual explanation, not a counterfactual search. ATEX-CF and C2Explainer edit pairwise graph edges and are not included in the comparison.

We report three metrics on non-isolated nodes. Success rate is how often the returned edit flips the prediction. Explanation size is how many incidences change. Sparsity is how much of the original incidence structure is left unchanged.

We compare bidirectional incidence search with deletion-only and graph-native baselines, then ask whether the returned edits use additions without larger explanations, how hop radius trades coverage for search noise, where the expansion fails, and how the additive-rate controls behave.

5.1 MAIN COMPARISON

Table 1 reports success, explanation size, and sparsity on the seven AllSet benchmarks. HyperEX and SHypX use the complementary-structure protocol in Section 5; they are factual reference points, not native counterfactual search. Among native methods, deletion-only CF-HyperGNNExplainer already outperforms star-expanded CF-GNNExplainer on every dataset where both run. CF-GNNExplainer cannot be evaluated on DBLP-CA because the expansion exceeds memory.

HyperCF raises success over that deletion baseline on six of seven datasets at radius two and five of seven at radius one. The mean explanation size does not increase on any dataset at either radius.

Table 1: Main performance comparison of proposed methods.

Metric	Method	Cora	Citeseer	Pubmed	Cora-CA	DBLP-CA	Mushroom	NTU2012
Success rate ↑	HyperEX	0.57	0.76	0.51	0.70	0.81	0.00	0.93
	SHypX	0.44	0.53	0.17	0.49	0.42	0.14	0.36
	CF-GNNExplainer	0.28	0.20	0.45	0.38	–	0.36	0.41
	CF-HyperGNNExplainer	0.89	0.91	0.70	0.85	**0.76**	0.60	0.95
	HyperCF (h=1)	0.90	0.92	0.89	0.90	0.75	0.93	0.95
	HyperCF (h=2)	**0.96**	**0.94**	**0.93**	**0.93**	0.60	**1.00**	**0.96**
Expl. size ↓	HyperEX	9.70	8.69	9.77	9.16	9.48	10.00	10.00
	SHypX	5.82	3.68	14.02	4.48	5.05	8.33	12.80
	CF-GNNExplainer	22.10	9.74	13.60	3.53	–	**1.20**	6.89
	CF-HyperGNNExplainer	2.40	1.90	7.18	1.69	2.16	4.36	4.19
	HyperCF (h=1)	2.20	1.81	6.90	1.64	2.04	3.98	3.93
	HyperCF (h=2)	**2.16**	**1.75**	**4.50**	**1.53**	**1.73**	3.10	**3.46**
Sparsity ↑	HyperEX	0.90	0.66	0.97	0.72	0.84	1.00	0.97
	SHypX	0.97	0.85	0.99	0.88	0.93	0.98	0.98
	CF-GNNExplainer	0.81	0.63	0.99	0.88	–	**1.00**	0.94
	CF-HyperGNNExplainer	**0.98**	**0.90**	**1.00**	0.93	**0.97**	**1.00**	**0.99**
	HyperCF (h=1)	**0.98**	0.89	**1.00**	**0.94**	**0.97**	**1.00**	**0.99**
	HyperCF (h=2)	**0.98**	0.89	**1.00**	0.92	0.96	0.97	0.97

On the three cocitation hypergraphs, deletion-only success is already high: 0.89 on Cora and 0.91 on Citeseer, and radius one adds only one percentage point on each. Radius two is where those cocitation gains appear: Cora rises to 0.96 and Citeseer to 0.94. Pubmed is the weak deletion case in this family. Deletion-only success is 0.70 at a mean size of 7.18; bidirectional search reaches 0.89 at radius one and 0.93 at radius two, with mean sizes 6.90 and 4.50. The two co-authorship datasets diverge. Cora-CA follows the cocitation pattern, from 0.85 to 0.90 and 0.93. On DBLP-CA, success falls from 0.76 to 0.75 at radius one and to 0.60 at radius two.

The remaining two datasets sit at opposite ends of the deletion baseline. Mushroom is the largest lift in the table, from 0.60 to 0.93 at radius one and 1.00 at radius two, with mean size falling from 4.36 to 3.10. NTU2012 is already near ceiling under deletion-only search at 0.95; radius two reaches 0.96. Sparsity stays comparable among the native counterfactual methods, so the extra successes are not obtained by denser incidence edits. The h=1 and h=2 rows are independent runs that differ only in the additive candidate set; Section 5.3 compares the two radii.

5.2 ROLE OF ADDITIVE EDITS

A smaller mean size with higher success does not by itself show that additions participate: the optimizer could still return only deletions, just fewer of them. Table 1 reports totals, so the split comes from the one-hop ablation in Table 6, which records deletions and additions on the seed-42 node set. After pruning, additions are 4.93 of 7.06 edits on Pubmed and 3.06 of 3.81 on Mushroom, the two datasets where deletion-only search is weakest, at 0.70 and 0.60. On Cora the retained mix is closer to balanced, with 1.21 deletions and 0.88 additions, consistent with a deletion baseline that already reaches 0.89.

This pattern matches the action-space claim. Deletion-only search fails on many Mushroom targets, and joining an existing attribute group is an edit that deletion search cannot express. Pubmed likewise starts from a larger deletion-only edit than the other cocitation datasets, and the bidirectional search

replaces part of that deletion load with additions. DBLP-CA is the reverse: the deletion baseline already succeeds on most nodes at a mean cost of 2.16, and expanding the search does not raise success, as discussed in Section 5.4.

5.3 CANDIDATE LOCALITY

Radius one and radius two search the same deletion set $\mathcal{D}(v_t)$ and differ only in $\mathcal{A}_h(v_t)$. With $h{=}1$ the target may join hyperedges of nodes in $\mathcal{N}_h(v_t)$; with $h{=}2$ it may also join hyperedges of nodes two incidence hops away. Both radii appear in Table 1 because the wider set is not uniformly better.

On Cora and Citeseer, radius one is nearly the deletion baseline, and the extra hop is what moves success, from 0.89 to 0.96 on Cora and from 0.91 to 0.94 on Citeseer. Pubmed and Mushroom already gain at radius one, from 0.70 to 0.89 and from 0.60 to 0.93; radius two mainly finishes remaining failures, reaching 0.93 and 1.00. Cora-CA improves at both radii. NTU2012 is insensitive to the extra hop, at 0.95 and then 0.96.

5.4 WHEN ADDITIVE SEARCH FAILS

DBLP-CA is the only dataset on which HyperCF falls below deletion-only search, from 0.76 to 0.75 at radius one and to 0.60 at radius two. The drop is not accompanied by a larger explanation: mean size falls from 2.16 under deletion-only search to 1.73 at radius two. A high deletion baseline does not explain it. On Cora that baseline is already 0.89, and radius two still raises success to 0.96. Co-authorship alone does not explain it either. Cora-CA is the same kind of hypergraph and is much smaller, with 2,708 nodes, 1,072 hyperedges, and a maximum hyperedge size of 43, against 41,302 nodes, 22,363 hyperedges, and a maximum of 202 on DBLP-CA. On Cora-CA, radius two raises success from 0.85 to 0.93.

The clearest measured difference accompanying the drop is the size of the additive candidate set. On DBLP-CA its mean size grows from 23 at radius one to 185 at radius two, the median from 12 to 83, and the maximum from 194 to 1,859, while success falls from 0.75 to 0.60. Radius one keeps the smaller set and stays within one percentage point of deletion-only search. Nodes are papers and hyperedges are authors. One possible explanation is that joining an existing author hyperedge often reconnects the target to papers already reachable through its current coauthors, limiting the new context introduced by additional candidates. The same pattern appears when additive updates are made more active. On a separate one-seed scan, raising the additive step ratio from 0.01 to 0.5 lowers DBLP-CA success from 75.1% to 71.5%, both below the deletion-only 76.2% on that split, whereas every tested ratio beats deletion-only search on Cora-CA. Moving the additive initialization from -4 to -1 lowers DBLP-CA success from 74.8% to 54.6%. The one-hop pruning ablation leaves DBLP-CA success unchanged and only removes surplus additions, suggesting that the missing flips are not caused by retaining a large explanation. The main table reports both radii.

6 CONCLUSION

Deletion-only incidence search cannot express a counterfactual that requires a target to join an existing hyperedge. HyperCF searches both directions on the incidence matrix, with a hop-bounded additive candidate set, learning-rate calibration based on deletion gradients with a coupled additive rate, and symmetric redundancy pruning.

On seven public benchmarks, this search raises success over deletion-only CF-HyperGNNExplainer on five datasets at radius one and six at radius two, and mean explanation size does not increase. The largest lifts are on Pubmed and Mushroom, where deletion is weakest and the retained edits contain more additions. Symmetric pruning removes surplus joins without changing success. DBLP-CA is the exception: widening the candidate set from radius one to two lowers success from 0.75 to 0.60, while the mean explanation size falls from 2.16 to 1.73.

A larger incidence action space therefore helps when nearby hyperedges carry unused context, and hurts when they do not. Adaptive candidate selection, joint incidence and feature edits, the creation of new hyperedges, and explanations for hyperedge- or hypergraph-level prediction remain open.

Reproducibility Statement

Anonymous code and configuration files will be included in the supplementary material. The datasets, AllSet split, base HGCN, sampling protocol, search budget, and metrics are specified in Section 5. The search and prune are given in Algorithm 1 and Algorithm 2.

AI Use Statement

We used generative AI tools to help retrieve related work, discuss method and experiment design, draft and revise manuscript prose, and polish wording. We did not use them to generate datasets, prove claims, or produce results. All AI-assisted text, citations, and technical suggestions were reviewed by the authors, who take responsibility for the final paper.

References

Xia an Bi, Sheng Luo, Siyu Jiang, Yu Wang, Zhaoxu Xing, and Luyun Xu. Explainable and programmable hypergraph convolutional network for imaging genetics data fusion. *Information Fusion*, 100:101950, 2023. doi: 10.1016/j.inffus.2023.101950.

Mohit Bajaj, Lingyang Chu, Zi Yu Xue, Jian Pei, Lanjun Wang, Peter Cho-Ho Lam, and Yong Zhang. Robust counterfactual explanations on graph neural networks. *arXiv preprint arXiv:2107.04086*, 2022.

Federico Battiston, Giulia Cencetti, Iacopo Iacopini, Vito Latora, Maxime Lucas, Alice Patania, Jean-Gabriel Young, and Giovanni Petri. Networks beyond pairwise interactions: Structure and dynamics. *Physics Reports*, 874:1–92, 2020. doi: 10.1016/j.physrep.2020.05.004.

Ziheng Chen, Fabrizio Silvestri, Jia Wang, He Zhu, and Hongshik Ahn. Relax: Reinforcement learning agent explainer for arbitrary predictive models. In *Proceedings of the 31st ACM International Conference on Information and Knowledge Management*, pp. 252–261, 2022. doi: 10.1145/3511808.3557429.

Ziheng Chen, Jin Huang, Fabrizio Silvestri, Yongfeng Zhang, Hongshik Ahn, and Gabriele Tolomei. Joint factual and counterfactual explanations for top-k gnn-based recommendations. *ACM Transactions on Recommender Systems*, 13(2):1–26, 2025. doi: 10.1145/3731683.

Eli Chien, Chao Pan, Jianhao Peng, and Olgica Milenkovic. You are allset: A multiset function framework for hypergraph neural networks. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=hpBTIv2uy_E.

Yifan Feng, Haoxuan You, Zizhao Zhang, Rongrong Ji, and Yue Gao. Hypergraph neural networks. In *Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence*, pp. 4238–4245, 2019. doi: 10.1609/aaai.v33i01.33013558.

Flavio Giorgi, Cesare Campagnano, Fabrizio Silvestri, and Gabriele Tolomei. Natural language counterfactual explanations for graphs using large language models. In *Proceedings of the 28th International Conference on Artificial Intelligence and Statistics*, pp. 3565–3573, 2025a.

Flavio Giorgi, Fabrizio Silvestri, and Gabriele Tolomei. Combinex: A unified counterfactual explainer for graph neural networks via node feature and structural perturbations. In *preprint*, 2025b. venue to be confirmed.

Sunwoo Kim, Soo Yong Lee, Yue Gao, Alessia Antelmi, Mirko Polato, and Kijung Shin. A survey on hypergraph neural networks: An in-depth and step-by-step guide. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 6534–6544, 2024. doi: 10.1145/3637528.3671457.

Ana Lucic, Harrie Oosterhuis, Hinda Haned, and Maarten de Rijke. Focus: Flexible optimizable counterfactual explanations for tree ensembles. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pp. 5313–5322, 2022a. doi: 10.1609/aaai.v36i5.20468.

Ana Lucic, Maartje A. ter Hoeve, Gabriele Tolomei, Maarten de Rijke, and Fabrizio Silvestri. Cf-gnnexplainer: Counterfactual explanations for graph neural networks. In *Proceedings of The 25th International Conference on Artificial Intelligence and Statistics*, volume 151 of *Proceedings of Machine Learning Research*, pp. 4499–4511, 2022b. URL https://proceedings.mlr.press/v151/lucic22a.html.

Dongsheng Luo, Wei Cheng, Dongkuan Xu, Wenchao Yu, Bo Zong, Haifeng Chen, and Xiang Zhang. Parameterized explainer for graph neural network. In *Advances in Neural Information Processing Systems*, volume 33, pp. 19620–19631, 2020.

Jiali Ma, Ichigaku Takigawa, and Akihiro Yamamoto. C2explainer: Customizable mask-based counterfactual explanation for graph neural networks. In *Proceedings of the 2025 ACM Conference on Fairness, Accountability, and Transparency*, 2025. doi: 10.1145/3715275.3732012. 13 pages.

Sepideh Maleki, Ehsan Hajiramezanali, Gabriele Scalia, Tommaso Biancalani, and Kangway V. Chuang. Learning to explain hypergraph neural networks. *arXiv preprint*, 2023.

Ramaravind K. Mothilal, Amit Sharma, and Chenhao Tan. Explaining machine learning classifiers through diverse counterfactual explanations. In *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*, pp. 607–617, 2020. doi: 10.1145/3351095.3372850.

Rafael Poyiadzi, Kacper Sokol, Raul Santos-Rodriguez, Tijl De Bie, and Peter Flach. Face: Feasible and actionable counterfactual explanations. In *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*, pp. 344–350, 2020. doi: 10.1145/3375627.3375850.

Shiye Su, Iulia Duta, Lucie Charlotte Magister, and Pietro Liò. Explaining hypergraph neural networks: From local explanations to global concepts. *arXiv preprint arXiv:2410.07764*, 2024.

Gabriele Tolomei and Fabrizio Silvestri. Generating actionable interpretations from ensembles of decision trees. *IEEE Transactions on Knowledge and Data Engineering*, 33(4):1540–1553, 2021. doi: 10.1109/TKDE.2019.2945326.

Gabriele Tolomei, Fabrizio Silvestri, Andrew Haines, and Mounia Lalmas. Interpretable predictions of tree-based ensembles via actionable feature tweaking. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 465–474, 2017. doi: 10.1145/3097983.3098039.

Fabiano Veglianti, Lorenzo Antonelli, and Gabriele Tolomei. Counterfactual explanations for hypergraph neural networks. *arXiv preprint arXiv:2602.04360*, 2026.

Sandra Wachter, Brent Mittelstadt, and Chris Russell. Counterfactual explanations without opening the black box: Automated decisions and the gdpr. *Harvard Journal of Law & Technology*, 31: 841–887, 2017.

Naganand Yadati, Madhav Nimishakavi, Prateek Yadav, Vikram Nitin, Anand Louis, and Partha Talukdar. Hypergcn: A new method of training graph convolutional networks on hypergraphs. In *Advances in Neural Information Processing Systems*, volume 32, 2019.

Rex Ying, Dylan Bourgeois, Jiaxuan You, Marinka Zitnik, and Jure Leskovec. Gnnexplainer: Generating explanations for graph neural networks. In *Advances in Neural Information Processing Systems*, volume 32, 2019.

Leisheng Yu, Yanxiao Cai, Minxing Zhang, and Xia Hu. Self-explaining hypergraph neural networks for diagnosis prediction. In *Proceedings of the Conference on Health, Inference, and Learning*, volume 287 of *Proceedings of Machine Learning Research*, pp. 908–924, 2025.

Yu Zhang, Sean Bin Yang, Arijit Khan, and Cuneyt Gurcan Akcora. Atex-cf: Attack-informed counterfactual explanations for graph neural networks. In *International Conference on Learning Representations*, 2026.

Dengyong Zhou, Jiayuan Huang, and Bernhard Schölkopf. Learning with hypergraphs: Clustering, classification, and embedding. In *Advances in Neural Information Processing Systems*, volume 19, 2006.

A COMMONLY USED NOTATIONS

Table 2: Notation for structural counterfactual search.

Symbol	Meaning
V, E	node set and hyperedge set
N, M	numbers of nodes and hyperedges
$\boldsymbol{H}$, $\boldsymbol{H}'$	original and edited binary incidence matrices
$\boldsymbol{H}(\boldsymbol{\pi})$	soft incidence matrix with candidate entries relaxed via s_e
$\boldsymbol{\pi}$	concatenated deletion and addition logits ($\boldsymbol{\pi} = [\boldsymbol{\pi}_D; \boldsymbol{\pi}_A]$)
$\boldsymbol{\pi}_D, \boldsymbol{\pi}_A$	logits of deletion and addition candidates
s_e	soft mask of candidate incidence, $s_e \in (0, 1)$
$\boldsymbol{X}$	node feature matrix
D, B	diagonal node-degree and hyperedge-degree matrices
$S(\boldsymbol{H})$	normalized hypergraph propagation operator
$f(\boldsymbol{H})_{v_t}$	predicted class of v_t under incidence $\boldsymbol{H}$
v_t	target node
$\hat{y}_t$	original prediction of the target node
$\mathcal{D}(v_t)$	existing incidences eligible for deletion
$\mathcal{A}(v_t)$	full set of absent incidences eligible for addition
$\mathcal{N}_h(v_t)$	h-hop incidence neighborhood of v_t
$\mathcal{A}_h(v_t)$	hop-bounded additive candidates within radius h
Δ^-, Δ^+	deleted and added incidence sets in an edit
$\mathcal{L}_{\text{pred}}, \mathcal{L}_{\text{dist}}$	prediction loss and distance loss
β	weight of the distance loss
η_D, η_A	deletion and addition learning rates
ρ	learning-rate ratio

B DATASET STATISTICS

Table 3: Dataset statistics.

Dataset	$\lvert V\rvert$	$\lvert E\rvert$	Features	Classes	Max $\lvert e\rvert$	Min $\lvert e\rvert$	Avg $\lvert e\rvert$
Cora	2,708	1,579	1,433	7	5	2	3.03
Citeseer	3,312	1,079	3,703	6	26	2	3.20
Pubmed	19,717	7,963	500	3	171	2	4.35
Cora-CA	2,708	1,072	1,433	7	43	2	4.28
DBLP-CA	41,302	22,363	1,425	6	202	2	4.45
Mushroom	8,124	298	22	2	1,808	1	136.31
NTU2012	2,012	2,012	100	67	5	5	5.00

C HYPERPARAMETER SENSITIVITY

Table 4 varies the additive step ratio ρ on a separate $1{,}000$-node run, with the checkpoint, sampled nodes, and remaining settings held fixed. On Cora-CA, every tested ρ beats the deletion-only run on

that split, which scores 73.8%, and the curve is flat from 77.6% at ρ=0.1 to 78.4% at ρ=0.5, with a slight peak of 78.5% at ρ=0.2. On DBLP-CA the same sweep moves in the opposite direction: 75.1% at ρ=0.01 down to 71.5% at ρ=0.5, all below the deletion-only 76.2%. Overly active additive updates therefore disturb a search that did not need them. No single ratio is best on both datasets. We use ρ=0.1, which keeps the Cora-CA gain and avoids the drop at 0.5 on DBLP-CA.

Table 5 varies the additive-logit initialization around the Method default -4, where $\mathrm{sigmoid}(-4.0) \approx 0.018$. Cora, Citeseer, and DBLP-CA are best at -4 and degrade as initialization becomes more active; DBLP-CA falls from 74.8% to 54.6% at -1. Pubmed peaks at -3 with 93.6% and collapses at -1 with 65.5%. Cora-CA is weakly peaked at -2. The default -4 is not uniformly maximal, but it avoids the DBLP-CA collapse and matches the deletion-only calibration, where additions start almost off.

Table 4: Sensitivity to the additive step ratio ρ.

Dataset	CF-HyperGNNExplainer	ρ=0.01	ρ=0.05	ρ=0.1	ρ=0.2	ρ=0.5
Cora-CA	73.8%	76.4%	77.5%	77.6%	78.5%	78.4%
DBLP-CA	76.2%	75.1%	74.7%	74.8%	73.5%	71.5%

Table 5: Sensitivity to additive-logit initialization.

Dataset	init= -4	init= -3	init= -2	init= -1
Cora	**92.7%**	90.4%	87.0%	84.3%
Citeseer	**90.8%**	90.4%	89.2%	85.7%
Pubmed	86.7%	**93.6%**	82.8%	65.5%
Cora-CA	87.7%	89.0%	**89.4%**	89.3%
DBLP-CA	**74.8%**	72.2%	65.8%	54.6%

D MINIMALITY AND MECHANISM ABLATION

The distance term averages the edit penalty over a large candidate set, so a mask near the mask threshold can keep deletions and additions that are not required for the flip. Table 6 isolates the symmetric prune on five datasets at radius one, using the seed-42 node set rather than the three-seed mean of Table 1. Success is identical with and without pruning on every dataset in the table, so the prune is not trading flips for sparsity.

Without pruning, the extra size is almost entirely additive. On Pubmed, mean size is 23.26, of which 21.10 are additions and 2.16 are deletions; after pruning the total is 7.06, of which 4.93 are additions and 2.13 are deletions. On Mushroom the unpruned total is 29.93, of which 27.57 are additions and 2.36 are deletions, and the pruned total is 3.81, of which 3.06 are additions and 0.75 are deletions. On Pubmed the deletion count stays near its unpruned value, 2.16 to 2.13; on Mushroom it falls from 2.36 to 0.75. In both cases the prune mainly removes surplus joins. Cora, Cora-CA, and DBLP-CA show the same shape at smaller scale: totals fall from 6.32 to 2.08, 3.22 to 1.64, and 4.47 to 2.06, again by cutting additions from 5.02 to 0.88, 1.99 to 0.52, and 2.85 to 0.62 while deletions stay near their unpruned values.

Table 6: Symmetric-pruning comparison on five representative AllSet datasets (h=1). Explanation size is reported as total (delete, add).

Metric	**Method**	**Cora**	**Pubmed**	**Cora-CA**	**DBLP-CA**	**Mushroom**
Success rate ↑	HyperCF w/o prune	0.93	0.88	0.89	0.74	0.90
	HyperCF	0.93	0.88	0.89	0.74	0.90
Expl. size ↓	HyperCF w/o prune	6.32 (1.29, 5.02)	23.26 (2.16, 21.10)	3.22 (1.24, 1.99)	4.47 (1.62, 2.85)	29.93 (2.36, 27.57)
	HyperCF	**2.08** (1.21, 0.88)	**7.06** (2.13, 4.93)	**1.64** (1.12, 0.52)	**2.06** (1.44, 0.62)	**3.81** (0.75, 3.06)